

オブジェクト指向プログラム仕様記述言語におけるブラウザ機能の拡張

3S-6

-FASET(6) 図式表現記述による開発支援ツール-

大木 幹雄

大嶋 高行

武内 利夫

日本電子計算(株)

1 はじめに

プログラム仕様記述/プログラミングを行うとき、ブラウザのもつ機能が果たす役割は大きい。Smalltalk-80(以後ST-80)では、ブラウザは大きく5つのサブビューから構成されており、それぞれクラス構成、クラス、メソッド等に関する情報を提供してくれる。しかし、サブビューを渡り歩く利用順序はあらかじめ決められていることから、ブラウザは単に検索・一覧する機能のみならず、プログラム設計、プログラミングの単位作業までを規定していると言える。

そこでオブジェクト指向プログラミングの設計プロセスを探る手がかりとして、ブラウザに着目し、設計/プログラミング操作に必要な情報をビジュアル化すると共に、仕様記述のプロセスをガイドする機能を「図式表現記述による開発支援ツールGraphtalk」のサブ機能として実現した。

2 ブラウザ機能拡張の基本概念

以下に述べるブラウザはST-80の"System Browser"を基盤にしている。ブラウザは、サブビュー別に見るとメニュー選択によるトップダウンな分類構造になっており、原理的には従来の構造化設計と対応している。

例えばclass categoryは、仮想データ/抽象機能モジュールに、実現上のデータ/機能モジュールはclass、詳細機能群の関連はmessage categoryに相当するように階層的な分類構造をもっている。このように規定されたブラウザを用いて設計者が仕様を記述するとき、抽象レベルを下げながら、次のような作業を同時に行っている。

- (1) システムの構成要素の洗いだし
- (2) 構成要素にもたせるべき属性や機能の分類
- (3) 分類された属性、機能の階層化

これらの作業はブラウザの利用操作に反映される。しかしながら、method textを除いたサブビューはリスト形式であり、かつclassビューを除いて(3)で示したような階層構造を陽にユーザに提示する機能はサポートしていない。また(1)から(3)の作業は、知識の整理・分類と類似のタスクであると考えられることができるが、エキスパートシステム構築支援ツールのもつような知識の整理プロ

セスを支援するような機能は提供していない。そこで、このような課題を解決するためにブラウジング機能を拡張した。

3 Graphtalkの機能概要

Graphtalkのもつ目的は、オブジェクト指向言語のもつモデル記述性の良さと、ビジュアルなユーザインタフェース作成の良さを加味して、オブジェクト指向概念で統一した開発環境を提供することを目指したものである。Graphtalkは、仕様設計からプログラミングまでをすべて図式により行う。ブラウザの図式表現や、ジェネリック・タスクの考え方を応用したプロセス・ガイダンスもこの考え方に沿って実現されている。Graphtalkの機能的な構成は図1の通りで、図式による設計仕様記述支援から、実行可能なST-80のソースコード生成までを行う。

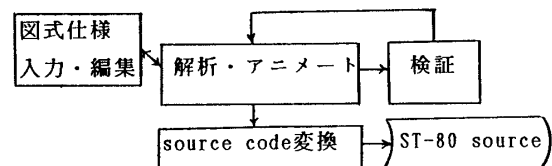


図1 Graphtalkの機能構成図

Graphtalkの仕様表現形式は、ST-80のオブジェクト指向概念を素直に表現できるようにしたもので、図2のような表現方法をとる。画面下部は図式から自動的に生成されたST-80ソースコードである。

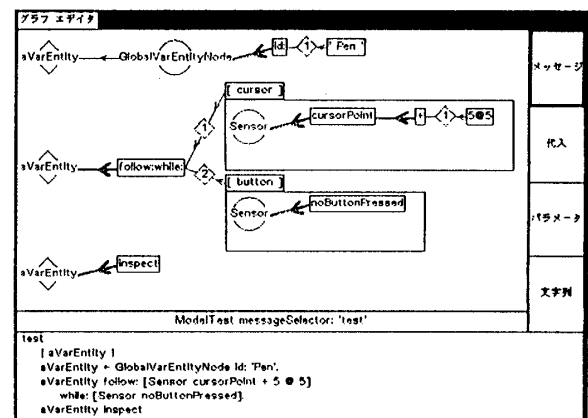


図2 Graphtalkの図式記述例

4 拡張したブラウザの表示及び操作

一般のCASEツールでは、構造図を図式表示し直接操作できるものが多い。ブラウザもclass階層については継承関係を把握するために図式表示することは既に行われている。class category, message category等についても、継承関係はないが、システムの構造を設計/プログラミング作業プロセスから決めていることから、図式的に表現し直接操作できなければならない。

そのようなことを可能にし図式表現したものが図3である。図3は仮想データ/抽象機能モジュール構造に対応するclass category、及び詳細機能に対応するmessage categoryがもつ階層的な分類構造を表したものである。message selectorにも分類階層化できる機能関係があるため、その関係を図式表示できるようにしている。

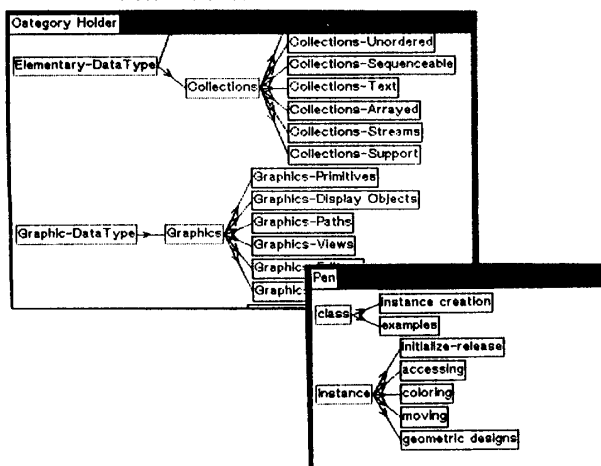


図3 操作するclass category, message category図

5 仕様記述プロセスのガイダンス機能との関連

ブラウザの内容を図3のように表示すると同時に、ブラウザ操作を仕様設計/プログラミング・プロセスの単位作業、並びにジェネリック・タスクにおける分類型知識整理プロセスと同等として捉え、そのプロセスも図式表現するようにした。

プロセスのガイダンス構造(階層構造をもつ)も図式を直接操作することが可能で、設計者の習熟度に従ってルールを追加したり、分類構造を変更することが容易に行えるようにしている。

具体的なプロセス・ガイダンス構造は図4に示す通りで、各ノード自体がオブジェクトとしてガイダンス・ルールを内部にもち、設計作業の局面、あるいは設計者からのリクエストに応じてアクティブな状態になる。するとガイダンス・ルールに従って質問を発し、ユーザの応答状況によって次に行うべき作業を判断しガイドする。

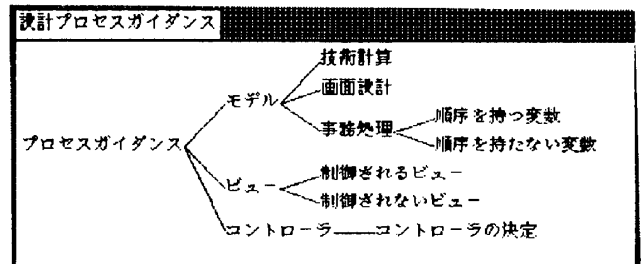


図4 仕様定義/プログラミング・ガイダンス図式

6 拡張ブラウザの評価

ブラウザの操作手順に着目してプロセスのガイダンスを行うことは、設計工程の自動化に向けての知識の蓄積につながる。そこで拡張したブラウザのうち、特にプロセスガイダンスについて考察・実験を繰り返した結果、次のような評価を得た。

- (1) 作業プロセスは概略的レベルでしか把握できない。
- (2) 典型的な作業パターンは、ごく少ない。
- (3) 設計の局面を質問応答形式で把握するには、質問の提示方法が重要である。すなわち、どのような質問を出すかがノウハウであり、経験を要する。
- (4) 作業プロセスの基本パターンとその変形とを識別する単位作業要素の洗いだしが必要であるが、それらをブラウザのサブビュー、ポップアップメニューのみから抽出するには、なお試行錯誤で追加しなければならない作業要素が多い。例えば、何のためにブラウザを複数、オープンするのか等が上げられる。

7 むすび

本稿で述べたGraphtalkは(財)情報処理振興事業協会の委託により協同システム開発(株)が推進している形式的仕様記述の環境開発を目指したF A S E Tプロジェクトの再委託により開発したCASEツールである。今後、成果の普及に向けて、他のCASEツールとのインタフェースを広げる予定である。

【参考文献】

- [1] 協同システム開発, 「ソフトウェア環境統合化技術開発計画-テクニカルレポート No.2」平成元年
- [2] G. Booch, "Object-Oriented Development", IEEE Trans. Software Eng., Vol SE-12, No. 2
- [3] B. Chandrasekaran, "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design", IEEE EXPERT Vol. 1, No. 3